On the Key Assertions (プログラムの基礎理論)

AUTHOR(S):

TAKASU, SATORU

CITATION:

TAKASU, SATORU. On the Key Assertions (プログラムの基礎理論). 数理解析研究所講究録 1973, 189: 13-37

ISSUE DATE:

1973-10

URL:

http://hdl.handle.net/2433/107224

数理解析研究所講究録
189巻 1973年 13-37

On the Key Assertions

Satoru Takasu (RIMS, Kyoto Univ.)

INTRODUCTION: Floyd's original approach [4] toward proving correctness of programs has been dependent upon the heuristic attachment of formulas between the statements of programs. So far this requirement has been necessary when we attach a formula to a loop so that this kind of attached formulas is called a key assertion.

In this paper, we try to determine the key assertions. First we reformulate the problem as to solve an equational system of Floyd's predicates [4] in which the strict verification conditions are considered to be equations. To solve such equational systems, we propose a method of substitution in which the logical equivalence is considered like an equality. Two examples are given and they showed that our method works well. However, in general it gives us the solutions expressed by formulas in infinitary logic and there exists a corelation with Engeler's results. [1]~[3], [10] On the other hand, if the function defined by a loop recursion is representable as a first-order formula within the background mathematical theory, then those formulas in infinitary logic reduce up to the formulas in first-order logic. In particular within the elementary number theory, our method determins the key assertions as first-order formulas. The above studies treate mainly the forward cases so that the key assertions determined are minimal in a sense. The backward cases will be studied elsewhere.

ACKNOWLEDGMENTS: The author expresses his hearty thanks to Dr. S. Igarashi and Dr. H. Ono for their kind discussions and comments.

1. Equational Systems of Floyd Predicates

1.1. Flowcharts in a Mathematical Theory

We assume throughout this paper that a mathematical theory T and a model M of T are given within the first order predicate logic.

A flowchart P in T consists of a finit number of function boxes, decision boxes and arrows connecting these boxes, provided that

(a) a function box contains an assignment statement v := f where v is a variable and f is a term defined in T and for each function box, there may be a finite number of incoming arrows and only one out-going arrow;

(b) a decision box contains a quantifier-free formula of T and for a decision box, there may be a finite number of incoming arrows and only two outgoing arrows to which the truth values of the formula with respect to M are preassigned; and

(c) we also assume that two boxes containing start and halt respectively are given and the former has only the outgoing arrow and the latter has only the incoming arrows.

Furthermore, for a flowchart P we assume that the vector x of input variables which never appear in the left part of any assignment statement, the vector y of program variables and the vector z of output variables are given and the flowchart is considered to compute a function

$$z = P(x): D_x \rightarrow D_z$$

where the input domain D_x and the output domain D_z are assumed

to be defined within the model M.

1.2. The Verification of a Flowchart P

Associated with the function $z = P(x)$: $D_x \rightarrow D_z$,two formulas $\varphi(x)$ and $\psi(x,z)$ are assumed to be given where $\varphi(x)$ expresses the input domain D_x and $\psi(x,z)$ indicates the desired input-output relation $z = P(x)$ over $D_x \times D_z$ when φ and ψ are interpreted within the model M.

We define that

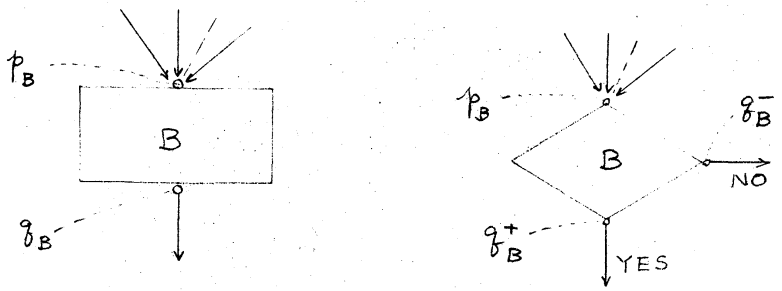
(i) P terminates over φ if for every input $\xi \in D_x$, namely $\varphi(\xi)$ is true, the computation of the flowchart terminates;

(ii) P is partially correct with respect to φ and ψ if for every ξ such that $\varphi(\xi)$ is true, if the computation terminates, then $\psi(\xi, P(\xi))$ is true; and

(iii) P is totally correct with respect to φ and ψ if for every ξ such that $\varphi(\xi)$ is true, the computation terminates and $\psi(\xi, P(\xi))$ is true.

1.3. Floyd Predicates

For each box B in a flowchart P, we attach one predicate $p_B(x,y)$ at the common end point of its incoming arrows and one predicate $q_B(x,y)$ at the initial point of its outgoing arrow when B is a function box and two predicates $q_B^+(x,y)$ and $q_B^-(x,y)$ when B is a decision box. The positions where predicates attached are shown in Fig.1.1. These predicates are called the F-predicates of the flowchart P.



A function box A decision box

Fig. 1.1. Attachment of F-predicates

1.4. Forward Substitutions

After the attachment of F-predicates to the flowchart P, we make the following identifications:

(1) First we establish the following domain-range relation for each box in the flowchart:

(1a) For each function box B containing an assignment statement of the form

$$y_i := f_B(x,y_1,\ldots,y_n)$$

we set the predicate $q_B(x,y_1,\ldots,y_n)$ to be the formula

$$(\exists y_i')\ p_B(x,y_1,\ldots,y_i',\ldots,y_n) \wedge y_i = f_B(x,y_1,\ldots,y_i',\ldots,y_n) .$$

(1b) For each decision box B containing a formula $t_B(x,y)$, the predicates $q_B^+(x,y)$ and $q_B^-(x,y)$ are set to be the formulas

$$p_B(x,y) \wedge t_B(x,y) \quad \text{and} \quad p_B(x,y) \wedge \neg t_B(x,y)$$

respectively.

(2) With respect to the flow of control, we introduce the following identifications among F-predicates:

(2a) When there exists only one arrow incoming to a box B and if it came from a box A, we identify q_A with p_B .

(2b) If a box B has more than one incoming arrows from $A_1,\ldots,$ A_k $(k\geq 2)$, we identify p_B with $q_{A_1}\vee q_{A_2}\vee\cdots\vee q_{A_k}$.

(2c) If B is the start-box of P, then we identify q_B with $\varphi(x)$.

1.5. Backward Substitutions

Similarly to the forward substitutions, we make the identifications replacing (1a) and (2c) by the followings respectively:

(1a') For each function box containing an assignment statement of the form

$$y_i := f_B(x,y_1,\ldots,y_n)$$

we set the predicate $p_B(x,y_1,\ldots,y_n)$ to be the formula

$$q_B(x,y_1,\ldots,f_B(x,y_1,\ldots,y_n),\ldots,y_n).$$

(2c') If B is the halt-box of P, then we identify $p_B(x,y)$ with the formula $\psi(x,z)$.

1.6. Equational Systems of F-predicates

Let $B_1, B_2, \ldots, B_k$ be the function boxes, $C_1, C_2,\ldots, C_h$ the decision boxes, B_0 the start-box of a given flowchart P, and

$(p_1,q_1),\ldots, (p_k,q_k)$;

$(p_{k+1},q_{k+1},q_{k+2}), \ldots, (p_{k+h},q_{k+2h-1},q_{k+2h})$

the F-predicates of $B_1,\ldots,B_k,C_1,\ldots,C_h$ respectively. The forward substitutions can be expressed as the following equqtional system of F-predicates:

$$q_0 \equiv \varphi(x)$$
$$p_1 \equiv L_1(q_{11},\ldots,q_{1r(1)})$$
$$q_1 \equiv M_1(p_1)$$
$$\ldots\ldots\ldots$$

$$p_k \equiv L_k(q_{k1},\ldots,q_{kr(k)})$$
$$q_k \equiv M_k(p_k)$$
$$p_{k+1} \equiv L_{k+1}(q_{k+11},\ldots,q_{k+1r(k+1)})$$
$$q_{k+1} \equiv M_{k+1}(p_{k+1})$$
$$q_{k+2} \equiv M_{k+2}(p_{k+1})$$
$$\ldots\ldots\ldots\ldots\ldots$$
$$p_{k+h} \equiv L_{k+h}(q_{k+h1},\ldots,q_{k+hr(k+h)})$$

where q_{ij} is one of $q_1,\ldots,q_{k+h}$. By the obvious substitutions we have the <u>forward equational system</u> EF_P <u>of F-predicates</u>:

$$q_1 \equiv N_1(\varphi)$$
$$q_2 \equiv N_2(q_{21},\ldots,q_{2r(2)})$$
$$\ldots\ldots\ldots\ldots\ldots\ldots$$
$$q_{k+2h} \equiv N_{k+2h}(q_{k+2h\ 1},\ldots,q_{k+2h\ r(k+2h)})$$

where N_i are the logical formulas depending on the contents of boxes and the flow of control.

Similary we obtain the backward equational system EB_P of F-predicates which contains $\psi(x,z)$ instead of $\varphi(x)$.

1.7. Utilization of the Equational System for the Verification of a Flowchart

To prove the partial correctness of a flowchart P with respect to φ and ψ, there need the following two steps:

(i) Determine the F-predicates $q_1,\ldots,q_{k+2h}$ as first order formulas satisfying EF_P or EB_P;

(ii) Prove the logical implication

$$q_E(x,y) \supset \psi(x,z) \qquad \text{(for the case of } EF_P\text{)}$$

or

$\varphi(x) \supset p_S(x,y)$ (for the case of EB_P)

where E is the box directly connected to the halt-box and S is the box directly connected from the start-box. These formulas will be called the implication formula of the system EF_P or EB_P respectively.

Our study will concern with the step (i) since the step (ii) is known to be done by some proof procedures such as the resolution principle when a solution of the step (i) was found.

2. Methods of Solutions

2.1. Method of Substitutions

Our equational system EF_P (or EB_P) concists of the first-order formulas involving the logical equivalences which are considered as a kind of equality. Therefore we may substitute the F-predicates by formulas as far as the latters are connected with the formers by logical equivalences provable at least in the model M under consideration. If we can connect each F-predicate to a formula without containning unknown F-predicates by this process of substitution, the latter formula is a <u>solution</u> of our equational system.

First, using the above process of substitution, we may reduce the number of F-predicates in the system EF_P (or EB_P) when we select the cutpoints on the arcs of the flowchart in such a way that every loop includes at least one such cut point and compose the functions of assignment statements between the cut points.

After this, there remains to determine the F-predicates attached to the loops, namely the key assertions, which are

minimal or maximal if we use EF_P or EB_P respectively. To determine the key assertions, the process of substitutions seems not to work by a first glance. However this will work well in the following examples where we introduce a technique making explicit the number of visit to the loop during the past computation or the future computation in the case of EF_P or EB_P respectively. For the general case, we shall introduce another method in the next section by which our method of substitutions is also justified.

In the following examples and in the next section, we shall use essentially the following concept known in mathematical logic: [8]

Definition: Let T be a mathematical theory and M a model of T. We assume that a correspondence between $m \in M$ and a term $\overline{m}$ in T is given (for example nutural number m and numeral $\overline{m} = 0^{(m)}$ where $0^{(m)}$ is the result of m fold applications of successor function to 0). A function $f(x_1,\ldots,x_n)$ defined within M is said to be representable [8] in T if there exists a well-formed formula $\Gamma(x_1,\ldots,x_n,x_{n+1})$ with n+1 free variables such that for any $(m_1,\ldots,m_n) \in M^n$,

(1) if $m_{n+1} = f(m_1,\ldots,m_{n+1})$ then $\vdash_T \Gamma(\overline{m}_1,\ldots,\overline{m}_{n+1})$, and

(2) $\vdash_T (\exists!\, x_{n+1})\, \Gamma(x_1,\ldots,x_{n+1})$.

Example 1. We consider the flowchart of Fig.2.1 which compute x !. To this flowchart we attach the predicate $\varphi(x) \equiv x > 0$, $p(x,y_1,y_2) \equiv y_1 = 1 \wedge y_2 = x$, $\psi(x,z) \equiv (z = x!)$ and $q(x,y_1,y_2)$ is considered to be unknown. Now we modify the flowchart of Fig.2.1 to the flowchart of Fig. 2.2 introducing an auxiliary variable i. We attach the predicates $\varphi(x)$, $\psi(x,z)$,

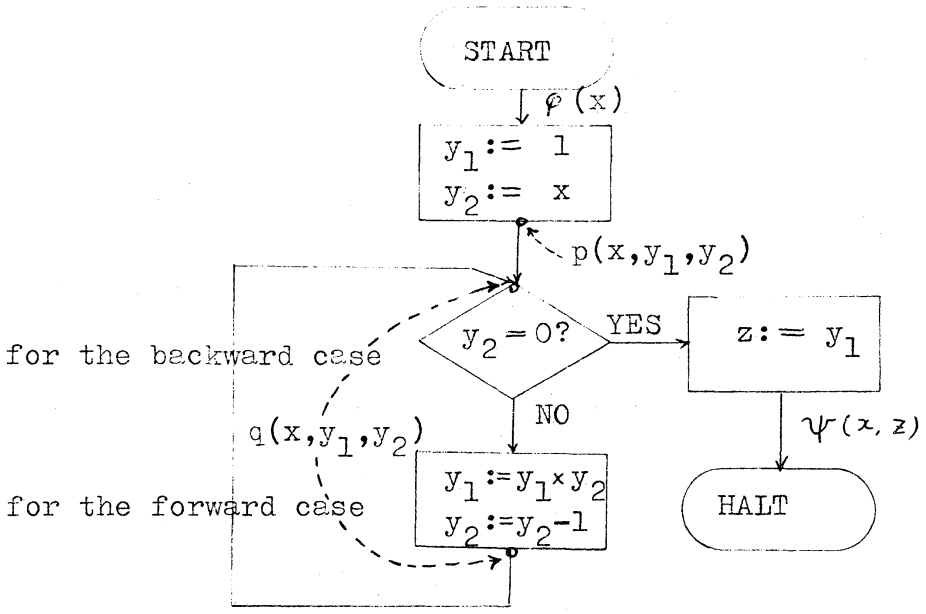


Fig.2.1. A flowchart to compute x!

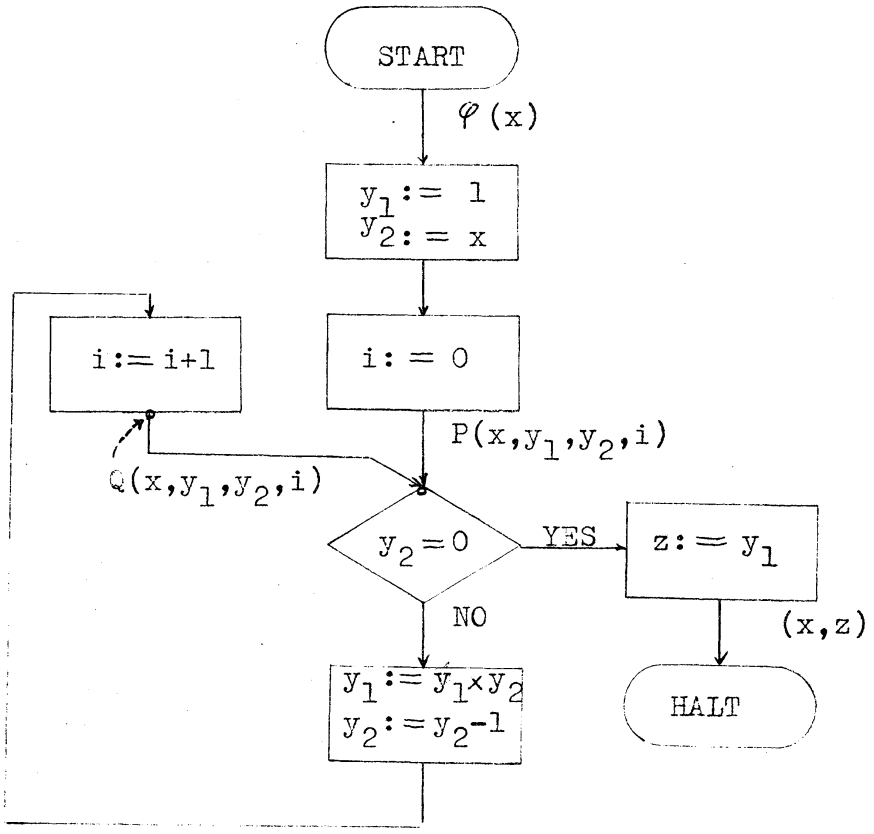


Fig.2.2. Modified flowchart

$P(x,y_1,y_2,i) \equiv y_1 = 1 \wedge y_2 = x \wedge i = 0$, and $Q(x,y_1,y_2,i)$ is to be determined. Note that the predicates q and Q are related by the formula

$$q(x,y_1,y_2) \equiv \bigvee_{i=1}^{\infty} Q(x,y_1,y_2,i)$$

where $\bigvee_{i=1}^{\infty}$ denotes the infinite disjunction. Therefore $q(x,y_1,y_2)$ is not a first-order formula but a formula in infinitary logic if the size of each determined formula $Q(x,y_1,y_2,i)$ depends on the variable i.

Now the system EF_P of the flowchart of Fig.2.2 becomes:

$$\begin{aligned}(1)\ Q(x,y_1,y_2,i) \equiv \exists j\, \exists z_1 \exists z_2 \\ (P(x,z_1,z_2,j) \vee Q(x,z_1,z_2,j)) \\ \wedge\ z_2 \neq 0 \wedge j = i-1 \wedge y_1 = z_1 \times z_2 \wedge y_2 = z_2 - 1\end{aligned}$$

and

$$(2)\ Q(x,y_1,y_2,i) \wedge y_2 = 0 \wedge z = y_1 \supset \psi(x,z).$$

As a background mathematical theory, we assume the elementary number theory in which $z = x!$ is representable by a first-order formula so that $z = x!$ can be treated as a first-order formula.

To determine $Q(x,y_1,y_2,i)$, we may rewrite (1) for each numeral $\overline{i} > 0$ as follows:

$$(3)\quad \begin{cases} Q(x,y_1^{(i)},y_2^{(i)},\overline{i}) \equiv \exists y_1^{(i-1)} \exists y_2^{(i-1)}\ Q(x,y_1^{(i-1)},y_2^{(i-1)},\overline{i-1}) \\ \qquad \wedge\ y_2^{(i-1)} \neq 0 \wedge y_1^{(i)} = y_1^{(i-1)} \times y_2^{(i-1)} \\ \qquad \wedge\ y_2^{(i)} = y_2^{(i-1)} - 1 \\ Q(x,y_1^{(i-1)},y_2^{(i-1)},\overline{i-1}) \equiv \exists y_1^{(i-2)} \exists y_2^{(i-2)} \cdots\cdots\cdots \\ \cdots\cdots\cdots\cdots\cdots\cdots \\ Q(x,y_1^{(1)},y_2^{(1)},1) \equiv \exists y_1^{(0)} \exists y_2^{(0)}\ P(x,y_1^{(0)},y_2^{(0)},0) \\ \qquad \wedge\ y_2^{(0)} \neq 0 \wedge y_1^{(1)} = y_1^{(0)} \times y_2^{(0)} \\ \qquad \wedge\ y_2^{(1)} = y_2^{(0)} - 1 \end{cases}$$

From the background mathematical theory we have

$$y_2^{(i-1)} = y_2^{(i)} + \bar{1}, \quad y_1^{(i-1)} = y_1^{(i)} / (y_2^{(i)} + \bar{1}) ,$$

$$y_2^{(i-2)} = y_2^{(i)} + \bar{2}, \quad y_1^{(i-2)} = y_1^{(i)} / (y_2^{(i)} + \bar{1})(y_2^{(i)} + \bar{2}),$$

$$\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots$$

$$y_2^{(0)} = y_2^{(i)} + \bar{i}, \quad y_1^{(0)} = y_1^{(i)} / (y_2^{(i)} + \bar{1})(y_2^{(i)} + \bar{2}) \cdots (y_2^{(i)} + \bar{i}) .$$

By the definition of the predicate P, we have $y_1^{(0)} = 1$ and $x = y_2^{(0)}$ so that there holds

$$y_1^{(i)} (y_2^{(i)}!) = x! .$$

Therefore if we make the successive substitutions to the first formula of (3) using the rest of formulas we have

$$Q(x, y_1^{(i)}, y_2^{(i)}, \bar{i}) \equiv (y_1^{(i)} \times (y_2^{(i)}!) = x!) \wedge \bar{i} = x - y_2^{(i)}$$

so that

$$\begin{aligned} q(x, y_1, y_2) &\equiv \bigvee_{j=1}^{\infty} (y_1 \times (y_2!) = x! \wedge \bar{j} = x - y_2) \\ &\equiv y_1 \times (y_2!) = x! \end{aligned}$$

since $\bar{j} = x - y_2$ is true for the unique value if x and y_2 are given. Now it is clear that the implication formula of EF_P holds.

We consider the following system EB_P of the flowchart of Fig.2.1. (This time Fig.2.2 is unnecessary.)

$$p(x, y_1, y_2) \wedge y_2 = 0 \supset \psi(x, z) \wedge z = y_2$$

$$p(x, y_1, y_2) \wedge y_2 \neq 0 \supset q(x, y_1, y_2)$$

$$q(x, y_1, y_2) \wedge y_2 = 0 \equiv \psi(x, z) \wedge z = y_2$$

$$q(x, y_1, y_2) \wedge y_2 \neq 0 \equiv q(x, y_1 \times y_2, y_2 - 1).$$

To determine $q(x, y_1, y_2)$, we utilize the process of substitution as follows:

$$\begin{aligned} q(x, y_1, y_2) &\equiv q(x, y_1, y_2) \wedge y_2 = 0 \vee q(x, y_1, y_2) \wedge y_2 \neq 0 \\ &\equiv \psi(x, y_1) \wedge y_2 = 0 \vee q(x, f(y_1, y_2), g(y_2)) \wedge y_2 \neq 0 \\ &\equiv \cdots\cdots\cdots\cdots\cdots\cdots \end{aligned}$$

$$\equiv \bigvee_{i=0}^{k}\Big\{\psi(x,f^{(i)}(y_1,y_2),g^{(i)}(y_2))\wedge g^{(i)}(y_2)=0$$

$$\wedge\ \Big(\bigwedge_{j=0}^{i} g^{(j)}(y_2)\neq 0\Big)$$

$$\vee\ q(x,f^{(k+1)}(y_1,y_2),\ g^{(k+1)}(y_2))\wedge \bigwedge_{j=0}^{k} g^{(j)}(y_2)\neq 0 \qquad \text{.....(4)}$$

where $f^{(i)}$ and $g^{(i)}$ are defined by setting

$$f(y_1,y_2)=y_1\times y_2,\quad g(y_2)=y_2-1$$

and

$$f^{(0)}(y_1,y_2)=y_1,$$
$$f^{(i)}(y_1,y_2)=f(f^{(i-1)}(y_1,y_2),g^{(i-1)}(y_2)),$$
$$g^{(0)}(y_2)=y_2\ ,$$
$$g^{(i)}(y_2)=g(g^{(i-1)}(y_2)).$$

We set

$$K(x,y_1,y_2)\equiv \bigvee_{i=0}^{\infty}\Big\{\psi(x,f^{(i)}(y_1,y_2),g^{(i)}(y_2))\wedge g^{(i)}(y_2)=0$$

$$\wedge\ \Big(\bigwedge_{j=0}^{i} g^{(j)}(y_2)\neq 0\Big)\Big\}\ .$$

From (4) it is clear that there holds

$$K(x,y_1,y_2)\supset q(x,y_1,y_2)\ .$$

On the other hand we have

$$q(x,y_1,y_2)\supset K(x,y_1,y_2)$$

in our model since $g^{(i)}(y_2)$ decreases when i increases. Hence we have

$$q(x,y_1,y_2)\equiv K(x,y_1,y_2)$$
$$\equiv y_1=x!\wedge y_2=0$$
$$\vee\ y_1\times y_2=x!\wedge y_2-1=0$$
$$\vee\ (y_1\times y_2)\times(y_2-1)=x!\wedge y_2-2=0$$
$$\cdots\cdots\quad\cdots\cdots\quad\cdots\cdots$$

$$\equiv \bigvee_{i=0}^{\infty} (y_1 \times (y_2!) = x! \wedge y_2 = i),$$

$$\equiv y_1 \times (y_2!) = x! \wedge (\bigvee_{i=0}^{\infty} y_2 = i),$$

$$\equiv y_1 \times (y_2!) = x! \ .$$

Therefore the minimal and maximal key assertions are the same for this case. Note also that the implication formulas of our system EB_P involving the infinite disjunction $K(x,y_1,y_2)$ clearly expresses the termination of our program.

Example 2. We consider the flowchart of Fig.2.3 which computes gcd(m,n). To this flowchart we attach the formulas:

$$\varphi(m,n) \equiv m>0 \wedge n>0,$$
$$p(m,n,c,d,r) \equiv c=m \wedge d=n \wedge \varphi(m,n)$$
$$\psi(m,n,d) \equiv d=gcd(m,n)$$

and the predicate $q(m,n,c,d,r)$ is to be determined. ($s(m,n,c,d,r)$ will be used when we consider EB_P.) We further parametrize the predicates p and q to the predicates P and Q respectively, introducing a control variable i which expresses the number of visits to the loop.

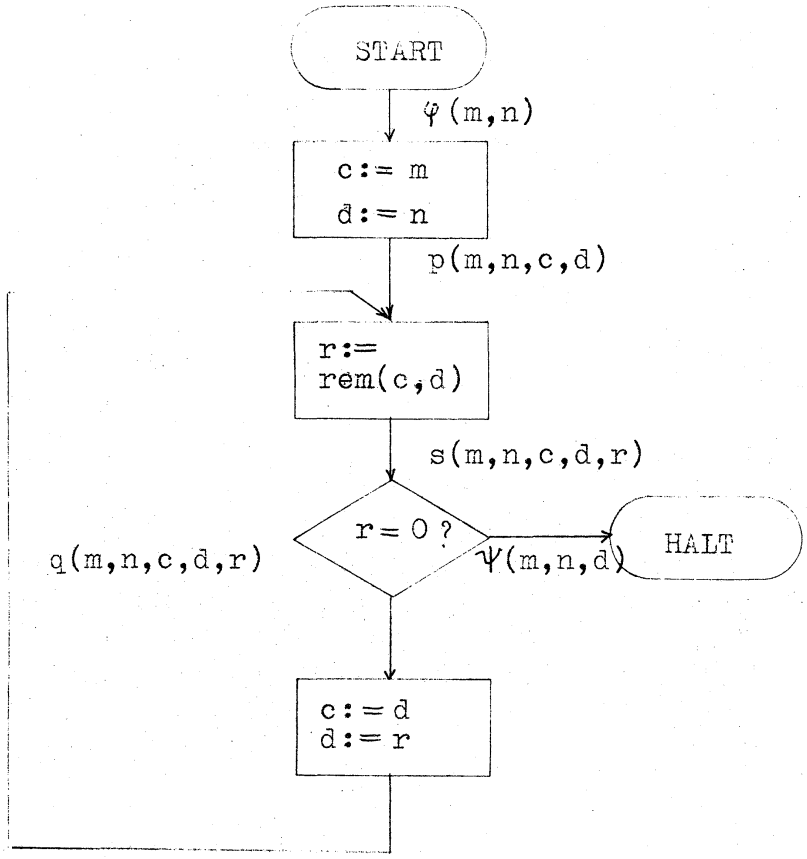


Fig.2.3. A flowchart to compute gcd(m,n)

We have the system EF_P as follows:

$$Q(x,y,0) \equiv c = m \wedge d = n \wedge i = 0 \wedge \varphi(m,n) \quad ,$$

$$Q(x,y,\overline{i}) \equiv \exists c' \exists d' \exists r'.\ Q(x,c',d',r',\overline{i-1}) \wedge r = rem(c,d)$$
$$\wedge\ c = d' \wedge d = r \quad ,$$

$$(\exists r'.\ Q(m,n,c,d,r',\overline{i}) \wedge r = rem(c,d) \wedge r = 0) \supset d = gcd(m,n)$$

where $x = (m,n)$ and $y = (c,d,r)$. For each numeral $\overline{i}$, we rewrite EF_P as follows:

$$Q(m,n,c^{(i)},d^{(i)},r^{(i)},\overline{i})$$
$$\equiv \exists c^{(i-1)} \exists d^{(i-1)} \exists r^{(i-1)}.\ Q(m,n,c^{(i-1)},d^{(i-1)},r^{(i-1)},\overline{i-1})$$
$$\wedge\ r^{(i)} = rem(c^{(i-1)},d^{(i-1)}) \wedge r^{(i)} \neq 0$$
$$\wedge\ c^{(i)} = d^{(i-1)} \wedge\ d^{(i)} = r^{(i)},$$
$$\cdots\cdots\cdots\cdots\cdots\cdots$$

$$Q(m,n,c^{(1)},d^{(1)},r^{(1)},1)$$
$$\equiv \exists c^{(0)} \exists d^{(0)} .\ c^{(0)} = m \wedge d^{(0)} = n \wedge m>0 \wedge n>0$$
$$\wedge\ r^{(1)} = \mathrm{rem}(c^{(0)},d^{(0)}) \wedge r^{(1)} \neq 0$$
$$\wedge\ c^{(1)} = d^{(0)} \wedge d^{(1)} = r^{(1)} .$$

If we set

$$a^{(0)} = 0,$$
$$b^{(0)} = 1,$$
$$\underline{a}^{(0)} = 1,$$
$$\underline{b}^{(0)} = 0,$$

and

$$a^{(i)} = \underline{a}^{(i-1)} - \mathrm{quo}(c^{(i-1)},d^{(i-1)}) \times a^{(i-1)},$$
$$b^{(i)} = \underline{b}^{(i-1)} - \mathrm{quo}(c^{(i-1)},d^{(i-1)}) \times b^{(i-1)},$$
$$\underline{a}^{(i)} = a^{(i-1)},$$
$$\underline{b}^{(i)} = b^{(i-1)},$$

then there hold

$$c^{(j)} = \underline{a}^{(j)} m + \underline{b}^{(j)} n \text{ , and}$$
$$d^{(j)} = a^{(j)} m + b^{(j)} n, \text{ where } j = 0,1,2,\ldots .$$

Since $a^{(j)}$, $b^{(j)}$, $\underline{a}^{(j)}$ and $\underline{b}^{(j)}$ are recursively defined functions of m, n, and j, they are representable in the elementary number theory so that we have

$$Q(m,n,c,d,r,i) \equiv c = \underline{a}^{(i)} m + \underline{b}^{(i)} n \wedge d = a^{(i)} m + b^{(i)} n$$
$$\wedge\ r = d \wedge r \neq 0 \wedge m>0 \wedge n>0$$

where i is a proper variable of the predicate Q. Therefore we have

$$q(m,n,c,d,r) \equiv \bigvee_{i=1}^{\infty} Q(m,n,c,d,r,i)$$
$$\equiv \exists i .\ i \geq 1 \wedge Q(m,n,c,d,r,i) .$$

Now it is clear that the implication formula $\hat{r}$ of $EF_p(q)$ or $EF_p(Q)$ holds if one knows the theorem

$$\exists a \exists b \exists \underline{a} \exists \underline{b} .\ c = \underline{a}m + \underline{b}n \wedge d = am + bn \wedge r = 0$$
$$\supset d = \gcd(m,n) .$$

We consider the system ED_P for the flowchart of Fig.2.3 where we use $s(m,n,c,d,r)$ instead of $q(m,n,c,d,r)$:

$$q(m,n,c,d,r) \supset s(m,n,c,d,rem(c,d)),$$

$$s(m,n,c,d,r) \wedge r \neq 0 \equiv s(m,n,c,d,rem(c,d)),$$

$$s(m,n,c,d,r) \wedge r = 0 \equiv \psi(m,n,d).$$

First we have the followings by the process of substitution:

$$s(m,n,c,d,r) \equiv s(m,n,c,d,r) \wedge r = 0 \vee s(m,n,c,d,r) \wedge r \neq 0,$$

$$\equiv \psi(m,n,d) \wedge r = 0 \vee s(m,n,f(c,d,r),g(c,d,r),h(c,d,r)),$$

$$\cdots\cdots\cdots\cdots\cdots\cdots\cdots$$

$$\equiv \bigvee_{i=0}^{k} \Big\{ \psi(m,n,g^{(i)}(c,d,r)) \wedge h^{(i)}(c,d,r) = 0 \wedge \Big(\bigwedge_{j=0}^{i-1} h^{(j)}(c,d,r) \neq 0 \Big) \Big\}$$

$$\vee \Big\{ s(m,n,f^{(k+1)}(c,d,r),g^{(k+1)}(c,d,r),h^{(k+1)}(c,d,r)) \wedge \Big(\bigwedge_{j=0}^{k} h^{(j)}(c,d,r) \neq 0 \Big) \Big\} \qquad \cdots\cdots (*)$$

where $f^{(i)}$, $g^{(i)}$ and $h^{(i)}$ are defined by setting

$$f(c,d,r) = d, \quad g(c,d,r) = r, \quad h(c,d,r) = rem(d,r),$$

and

$$f^{(0)}(c,d,r) = c, \quad g^{(0)}(c,d,r) = d, \quad h^{(0)}(c,d,r) = r,$$

$$f^{(i)}(c,d,r) = f(f^{(i-1)}(c,d,r),g^{(i-1)}(c,d,r),h^{(i-1)}(c,d,r)),$$

$$g^{(i)}(c,d,r) = g(f^{(i-1)}(c,d,r),g^{(i-1)}(c,d,r),h^{(i-1)}(c,d,r)),$$

$$h^{(i)}(c,d,r) = h(f^{(i-1)}(c,d,r),g^{(i-1)}(c,d,r),h^{(i-1)}(c,d,r)).$$

If we make explicit the operations, then the previously defined functions $a^{(j)}$, $b^{(j)}$, $\underline{a}^{(j)}$ and $\underline{b}^{(j)}$ where their arguments are c and d this time instead of m and n respectively, define the above functions, namely we have

$$f^{(j)}(c,d,r) = \underline{a}^{(j)}\ c + \underline{b}^{(j)}\ d\ ,$$
$$g^{(j)}(c,d,r) = a^{(j)}\ c + b^{(j)}\ d\ ,$$
$$h^{(j)}(c,d,r) = a^{(j+1)}\ c + b^{(j+1)}\ d.$$

Therefore the suffix j can be considered as a proper variable of predicates.

Now we set

$$K(m,n,c,d,r) \equiv \bigvee_{i=0}^{\infty} \Big\{ \psi(m,n,g^{(i)}(c,d,r)) \wedge h^{(i)}(c,d,r)=0 \wedge \Big(\bigwedge_{j=0}^{i-1} h^{(j)}(c,d,r) \neq 0 \Big) \Big\} .$$

Then from (*) we have $K(m,n,c,d,r) \supset s(m,n,c,d,r)$. On the other hand

$$\begin{aligned} g^{(1)} &= \mathrm{rem}(c,d) < g^{(0)} = d, \\ g^{(j+1)} &= a^{(j+1)}\ c + b^{(j+1)}\ d \\ &= (\underline{a}^{(j)} - \mathrm{quo}(f^{(j)},g^{(j)})a^{(j)})c \\ &\quad + (\underline{b}^{(j)} - \mathrm{quo}(f^{(j)},g^{(j)})b^{(j)})d \\ &= \underline{a}^{(j)}\ c + \underline{b}^{(j)} d - \mathrm{quo}(f^{(j)},g^{(j)})(a^{(j)}c+b^{(j)}d) \\ &= f^{(j)} - \mathrm{quo}(f^{(j)},g^{(j)})g^{(j)} \\ &= \mathrm{rem}(f^{(j)},g^{(j)}) < g^{(j)}, \end{aligned}$$

namely,

$$d = g^{(0)} > g^{(1)} > \ldots > g^{(j)} > \ldots \geq 0$$

and therefore

$$s(m,n,c,d,r) \supset \exists j.\ h^{(j)}(c,d,r) = 0$$

so that there holds

$$s(m,n,c,d,r) \supset K(m,n,c,d,r).$$

Hence we have

$$s(m,n,c,d,r) \equiv K(m,n,c,d,r) .$$

Using the properties of gcd we have

$$\begin{aligned}&\Psi(m,n,g^{(i)}(c,d,r))\wedge h^{(i)}(c,d,r)=0\\&\equiv \gcd(f^{(i)}(c,d,r),g^{(i)}(c,d,r)) = g^{(i)}(c,d,r)\\&\quad\wedge g^{(i)}(c,d,r)=\gcd(m,n)\wedge h^{(i)}(c,d,r)=0\\&\equiv \gcd(c,d)=\gcd(m,n)\wedge g^{(i)}(c,d,r)=\gcd(m,n)\\&\quad\wedge h^{(i)}(c,d,r)=0\end{aligned}$$

and

$$\begin{aligned}&\bigwedge_{j=0}^{i-1} h^{(j)}(c,d,r)\neq 0\\&\quad\equiv \forall j.\ j<i\supset h^{(j)}(c,d,r)\neq 0\end{aligned}$$

so that we conclude

$$\begin{aligned}&s(m,n,c,d,r)\equiv \gcd(m,n)=\gcd(c,d)\\&\quad\wedge \exists i.\ g^{(i)}(c,d,r)=\gcd(m,n)\wedge h^{(i)}(c,d,r)=0\\&\quad\wedge \forall j.\ j<i\supset h^{(j)}(c,d,r)\neq 0\ .\end{aligned}$$

Here we note that the implication formula of EB_P clearly describes the termination of P.

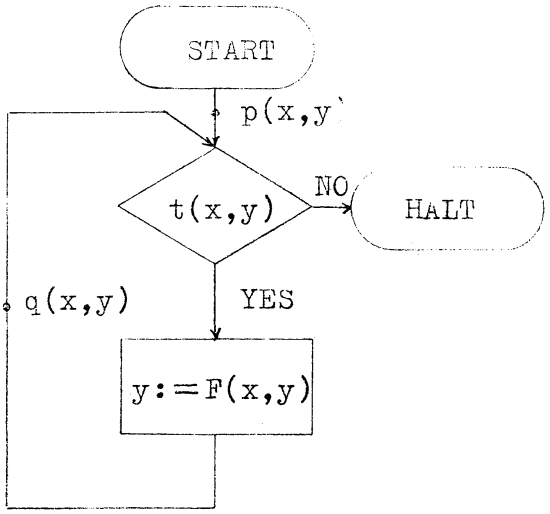
2.2. A General Consideration for the Forward Cases

It is known that[5] for any given flowchart there exists an equivalent flowchart having a single loop. Therefore we consider the typical flowcharts (a) and (b) of Fig.2.4, where predicates p(x,y) and q(x,y) are attached and q(x,y) is to be determined. We transform these flowcharts to the flowcharts (a) and (b) of Fig.2.5, introducing new integer variable i to express the number of visits to the loop.**) The predicates P(x,y,i) and Q(x,y,i) correspond to the predicates p(x,y) and q(x,y) respectively where they are related by the formulas

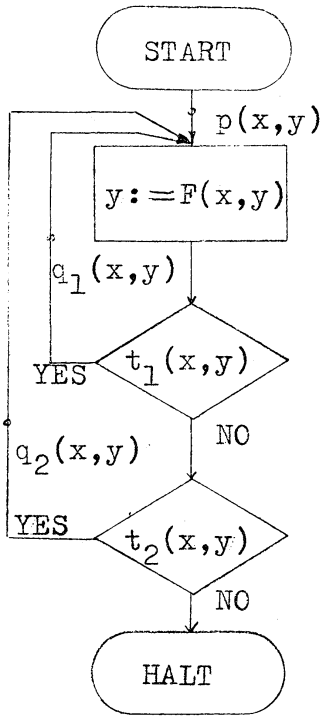
$$P(x,y,i)\equiv p(x,y)\wedge i=0$$

and

**) This means that our underlying logic is the many sorted logic.

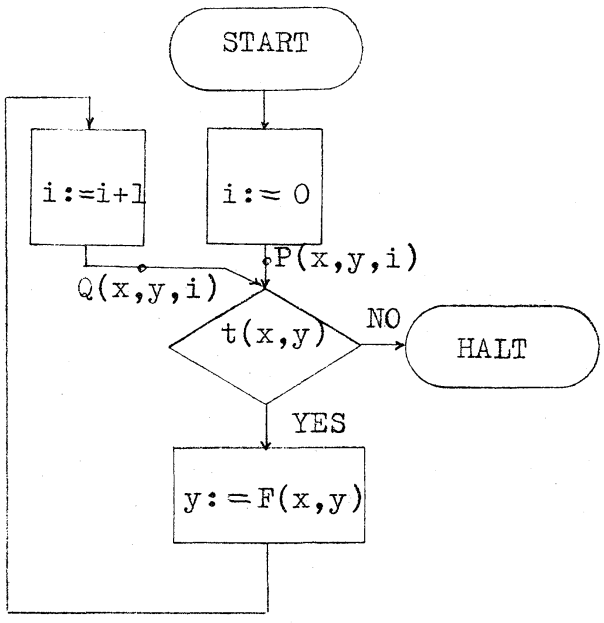


(a)

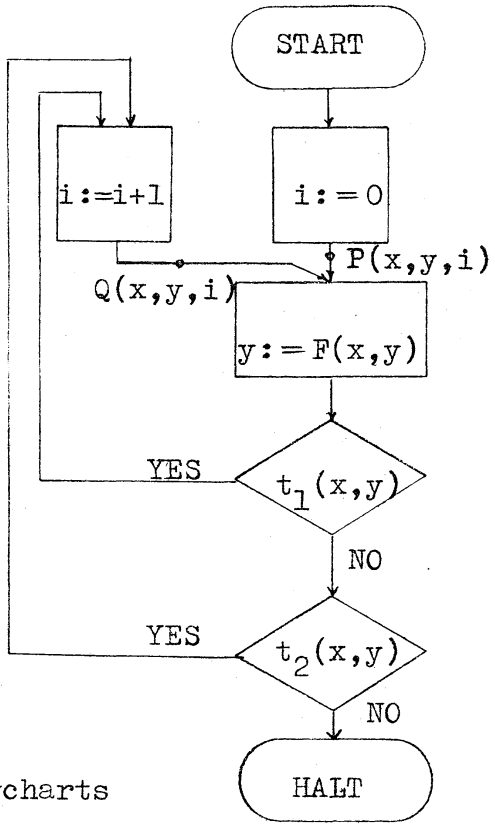


(b)

Fig.2.4. Typical Flowcharts



(a)



(b)

Fig.2.5. Augmented Typical Flowcharts

$$q(x,y) \equiv \bigvee_{i=1}^{\infty} Q(x,y,i).$$

Now our equational systems EF_p for the flowcharts of Fig.2.5 are given as follows:

(a) $P(x,y,i) \equiv p(x,y) \wedge i = 0$

$$Q(x,y,i) \equiv \exists z.(P(x,z,i-1) \vee Q(x,z,i-1)) \wedge y = F(x,z) \wedge t(x,z)$$

(b) $P(x,y,i) \equiv p(x,y) \wedge i = 0$

$$Q(x,y,i) \equiv \exists z.(P(x,z,i-1) \vee Q(x,z,i-1)) \wedge y = F(x,z) \wedge (t_1(x,y) \vee (\neg t_1(x,y) \wedge t_2(x,y))).$$

The above systems can be rewritten for each numeral m as follows:

(a),(b):

$$q(x,y,m) \equiv \exists z_1.Q(x,z_1,m-1) \wedge K(x,y,z_1),$$
$$Q(x,z_1,m-1) \equiv \exists z_2.Q(x,z_2,m-2) \wedge K(x,y,z_2),$$
$$\cdots\cdots\cdots \quad \cdots\cdots\cdots \quad \cdots\cdots\cdots$$
$$Q(x,z_{m-1},1) \equiv \exists z_m.P(x,z_m,0) \wedge K(x,y,z_m),$$

where

(a) $K(x,y,z) \equiv y = F(x,z) \wedge t(x,z)$, and

(b) $K(x,y,z) \equiv y = F(x,z) \wedge (t_1(x,y) \vee (\neg t_1(x,y) \wedge t_2(x,y)))$.

Therefore the process of substitution gives us the solution:

(a), (b):

$$q(x,y,m) \equiv \exists z_1 \exists z_2 \cdots\cdots \exists z_m.\ (K(x,y,z_1) \wedge K(x,z_1,z_2) \wedge \cdots \wedge K(x,z_{m-1},z_m) \wedge P(x,z_m,0))$$

where we note that for each numeral m the right hand side of $\equiv$ is a well-formed formula whose size is depending on m. The last fact implies that the variable i of $Q(x,y,i)$ is improper, namely it expresses, in general, the numbering of the sequence of different formulas so that $Q_i(x,y)$ is, in general, the proper notation of $Q(x,y,i)$.

By these observation a problem arises, namely "Under what condition does the parameter i become a proper variable of the predicate Q ? " . To examine this problem, we rewrite the solution as follows (we consider only the case (a) since the case (b) can be treated similarly):

$$Q(x,y,m) \equiv \exists z_1 \exists z_2 \ldots \exists z_m .$$
$$(y = F(x,z_1) \wedge z_1 = F(x,z_2) \wedge \ldots \wedge z_{m-1} = F(x,z_m))$$
$$\wedge (t(x,z_1) \wedge t(x,z_2) \wedge \ldots \wedge t(x,z_m)) .$$

We define G(x,y,m) by setting

$$G(x,y,0) = y$$
$$G(x,y,m) = F(x,G(x,y,m-1)),\ m = 1,\ 2,\ 3, \ldots$$

and we call it the function defined by a <u>loop-recursion</u>. Now we assume that this function G is representable in the background mathematical theory T and let $\Gamma(x,y,i,z)$ be the representing first-order formula. Under this assumption and at least in the model M of T, there holds:

$$Q(x,y,i) \equiv \exists z . \Gamma(x,z,i,y) \wedge \forall j . (j < i \supset t(x,G(x,z,j))$$

so that i is a proper variable of Q. Note that in this case we may write

$$q(x,y) \equiv \exists i .\ Q(x,y,i).$$

The function G defined by a loop-recursion is always representable within the elementary number theory T, provided that the one-step function F is representable in T. Therefore for the programs computing recursive functions, the minimal key assertions can be determined as first-order formulas.

2.3. Uniqueness and Beth's Definability (Forward Cases)

Returning to the general setting, we have the problem whether our solution is unique or not up to the provable (in M) logical equivalences. Before get into this problem, we mention the following well-known theorem in mathematical logic:

Beth's Definability Theorem:[9] Let $\mathcal{A}$ be a set of closed formulas, Q an n-ary predicate symbol of language($\mathcal{A}$) and $\mathcal{A}'$ the set of formulas obtained by substituting for Q in each formula of $\mathcal{A}$ an n-ary predicate symbol Q' which does not occur in language($\mathcal{A}$). Then if $Q(x_1,\ldots,x_n) \supset Q'(x_1,\ldots,x_n)$ is a consequence of $\mathcal{A} \cup \mathcal{A}'$, there is a formula Φ such that language(Φ) $\subseteq$ language($\mathcal{A}$), $Q \notin$ language(Φ) and $\forall x_1 \forall x_2 \ldots \forall x_n .\ Q(x_1,\ldots,x_n) \equiv \Phi$ is a consequence of $\mathcal{A}$.

It is also known[9] that the formula Φ can be explicitly constructed from a proof schema of the sequent

$$A \wedge Q(x_1,\ldots,x_n) \rightarrow A' \supset Q'(x_1,\ldots,x_n)$$

where A is the conjunction of the formulas in $\mathcal{A}$, A' is the conjunction of the formulas in $\mathcal{A}'$ and the proof schema itself is constructed within the symmetric Gentzen system.

Now going back to our system EF_p of Fig.2.5, we prove the sequent

$$A,\ Q_m(a,b) \rightarrow Q'_m(a,b)\ ,\ \vdash A'$$

where we set

$$A = \bigwedge_{i=0}^{\infty} \forall x \forall y .\ Q_i(x,y) \equiv \exists z .\ Y_{i-1}(x,y,z),$$

$$Y_{i-1}(x,y,z) = Q_{i-1}(x,z) \wedge X(x,y,z),$$

$$\vdash A' = \bigvee_{i=0}^{\infty} \exists x \exists y .\ Q'_i(x,z) \not\equiv \exists z .\ Y'_{i-1}(x,y,z),$$

$$Y'_{i-1}(x,y,z) = Q'_{i-1}(x,z) \wedge X(x,y,z)$$

$$Q_0(x,y) = P(x,y,0) = Q'_0(x,y).$$

$$\frac{Q_0(a,\beta_n),\ X(a,\beta_{n-1},\beta_n)\rightarrow \left\{\begin{array}{l} Q_0(a,\beta_n) \\ X(a,\beta_{n-1},\beta_n)\end{array}\right\}^{AND}}{A,\ Y_0(a,\beta_{n-1},\beta_n)\ \rightarrow\ Y_0'(a,\beta_{n-1},\beta_n),\ \neg A'}$$

$$\frac{A,\ Y_0(a,\beta_{n-1},\beta_n)\ \rightarrow\ \exists z\, Y_0'(a,\beta_{n-1},z),\ \neg A'}{A,\ \exists z\, Y_0(a,\beta_{n-1},z)\rightarrow \exists z\, Y_0'(a,\beta_{n-1},z),\ \neg A'}$$

$$\frac{A,\ \exists z\, Y_0(a,\beta_{n-1},z)\rightarrow Q_1'(a,\beta_{n-1})\left\{\begin{array}{l} Q_1'(a,\beta_{n-1}),\exists z\, Y_0'(a,\beta_{n-1},z) \\ \neg Q_1'(a,\beta_{n-1}),\neg\exists z\, Y_0'(a,\beta_{n-1},z)\end{array}\right\}^{AND},\ \neg A'}{A,\ \left\{\begin{array}{l}\exists z\, Y_0(a,\beta_{n-1},z),\ Q_1(a,\beta_{n-1}) \\ \neg\exists z\, Y_0(a,\beta_{n-1},z),\ \neg Q_1(a,\beta_{n-1})\end{array}\right\}^{OR} Q_1(a,\beta_{n-1})\rightarrow Q_1'(a,\beta_{n-1}),\ Q_1'(a,\beta_{n-1})\neq\exists z\, Y_0'(a,\beta_{n-1},z),\ \neg A'}$$

$$\frac{A,\ Q(a,\beta_{n-1})\equiv\exists z\, Y_0(a,\beta_{n-1},z),\ Q_1(a,\beta_{n-1})\rightarrow Q_1'(a,\beta_{n-1}),\ Q_1'(a,\beta_{n-1})\neq\exists z\, Y_0'(a,\beta_{n-1},z),\ \neg A'}{A,\ Q_1(a,\beta_{n-1})\ \rightarrow\ Q_1'(a,\beta_{n-1}),\ \neg A'}$$

$$\frac{A,\ Q_1(a,\beta_{n-1}),\ \ X(a,\beta_{n-2},\beta_{n-1})\rightarrow\left\{\begin{array}{l} Q_1'(a,\beta_{n-1}) \\ X(a,\beta_{n-2},\beta_{n-1})\end{array}\right\}^{AND},\ \neg A'}{}$$

- - - - - - - - - - - - - - - - - -

$$\frac{A,\ Q_{m-1}(a,\beta_1),\ X(a,b,\beta_1)\rightarrow\left\{\begin{array}{l} Q_{m-1}'(a,\beta_1) \\ X(a,b,\beta_1)\end{array}\right\}^{AND},\ \neg A'}{A,\ Y_{m-1}(a,b,\beta_1)\ \rightarrow\ Y_{m-1}'(a,b,\beta_1),\ \neg A'}$$

$$\frac{A,\ Y_{m-1}(a,b,\beta_1)\ \rightarrow\ \exists z\, Y_{m-1}'(a,b,z),\ \neg A'}{A,\ \exists z\, Y_{m-1}(a,b,z)\ \rightarrow\ \exists z\, Y_{m-1}'(a,b,z),\ \neg A'}$$

$$\frac{A,\ \exists z\, Y_{m-1}(a,b,z)\rightarrow Q_m'(a,b),\left\{\begin{array}{l} Q_m'(a,b),\exists z\, Y_{m-1}'(a,b,z) \\ \neg Q_m'(a,b),\ \neg\exists z\, Y_{m-1}'(a,b,z)\end{array}\right\}^{AND},\ \neg A'}{A,\ \left\{\begin{array}{l}\exists z\, Y_{m-1}(a,b,z),\ Q_m(a,b) \\ \neg\exists z\, Y_{m-1}(a,b,z),\ \neg Q_m(a,b)\end{array}\right\}^{OR},\ Q_m(a,b)\rightarrow Q_m'(a,b),\ Q_m'(a,b)\neq\exists z\, Y_{m-1}'(a,b,z),\ \neg A'}$$

$$\frac{A,\ Q_m(a,b)\equiv\exists z\, Y_{m-1}(a,b,z),\ Q_m(a,b)\rightarrow Q_m'(a,b),\ Q_m'(a,b)\neq\exists z\, Y_{m-1}'(a,b,z),\ \neg A'}{A,\ Q_m(a,b)\equiv\exists z\, Y_{m-1}(a,b,z),\ Q_m(a,b)\rightarrow Q_m'(a,b),\ \exists x\exists y\, Q_m'(x,y)\neq\exists z\, Y_{m-1}'(x,y,z),\ \neg A'}$$

$$\frac{A,\ \forall x\forall y\, Q_m(x,y)\equiv\exists z\, Y_{m-1}(x,y,z),\ Q_m(a,b)\rightarrow Q_m'(a,b),\ \exists x\exists y\, Q_m'(x,y)\neq\exists z\, Y_{m-1}'(x,y,z),\ \neg A'}{A,\ \forall x\forall y\, Q_m(x,y)\equiv\exists z\, Y_{m-1}(x,y,z),\ Q_m(a,b)\rightarrow Q_m'(a,b),\ \bigvee_i\exists x\exists y\, Q_i'(x,y)\neq\exists z\, Y_{i-1}'(x,y,z),\ \neg A'}$$

$$\frac{\bigwedge_i\forall x\forall y\, Q_m(x,y)\equiv\exists z\, Y_{m-1}(x,y,z),\ Q_m(a,b)\rightarrow Q_m'(a,b),\ \bigvee_i\exists x\exists y\, Q_i'(x,y)\neq\exists z\, Y_{i-1}'(x,y,z)}{A,\ Q_m(a,b)\ \rightarrow\ Q_m'(a,b),\ \neg A'}$$

Fig.2.6. Proof schema to prove the uniqueness

The proof schema of the above sequent is shown in Fig.2.6. Here we note that the treatements of $\bigwedge_i$ and $\bigvee_i$ are similar to that of $\forall i$ and $\exists i$ respectively. From the proof schema we read out that the solution constructed from the schema is identical with the solution obtained by the process of substitution and that size of the proof schema itself depends on the number m so that the size of $Q_m(x,y)$ depends on m in general.

Summerizing the aboves we have:

Theorem 1. For a given flowchart, there exists an equivalent flowchart for which the minimal key assertion can be determined as an infinite disjunction of first-order formulas and these first-order formulas satisfying the strict verification conditions are unique up to the logical equivalences.

Theorem 2. If the function defined by a loop recursion is representable in the background mathematical theory T, then the minimal key assertion of the loop can be determined as a first-order formula. In particular, if T is the elementary number theory, then every minimal key assertion of a program can be determined as a first-order formula provided that each function box computes a recursive function.

BIBLIOGRAPHY

[1] Engeler, E. : Algorithmic Properties of Structures, Math. Systems Theory 1(1967)183-195.

[2] Engeler, E. : Formal Languages: Automata and Structures, 81pp., Markham,1968.

[3] Engeler, E. : Structure and Meanings of Programs, Symposium on Semantics of Algorithmic Languages, Springer-Verlag, pp89-101 ,1971.

[4] Floyd, R.W. : Assigning Meanings to Programs, in Proc. Sym. in Applied Math. 19, Mathematical Aspects of Computer Sciences (Schwartz, J.T. ed.), AMS,(1967)pp.19-32.

[5] Hirose, K. and Oya, M. : General Theory of Flowcharts, U.S.-Japan Computer Conference Proceedings, (1972)367-371.

[6] Manna, Z. : The Correctness of Programs, J. of Computer and Systems Sciences, 3(1969)119-127.

[7] Manna, Z. : Properties of Programs and the First-Order Predicate Calculus, Jour. ACM, 16(1969)244-255.

[8] Mendelson, E. : Introduction to Mathematical Logic, Van Nostrand, 1964.

[9] Smullyan, R.M. : First-Order Logic, Springer-Verlag, 1968.

[10] Engeler,E. : Algorithmic Approximations, Jour. of Computer and System Sciences 5(1971)67-82.